

```
def getMean(num):
    if len(num) == 0:
        return 0
    else:
        cur_sum = 0
        for i in num:
            cur_sum = cur_sum+i
        return cur_sum/len(num)

def getMode(num):
    # we will use a tuple
    max_count = (0,0)
    # max_count[0] -> will give the total occurence of the specific number
    # max_count[1] -> will give the number itself
    for i in num:
        occurrences = num.count(i)
        if occurrences > max_count[0]:
            max_count = (occurrences,i)
    return max_count[1]
```

+ Code

+ Text

```
def GetMedian(num):
    # we need to first sort the list of numbers
    num.sort()

    if len(num) % 2 != 0:
        middle_index = int((len(num)-1)/2)
        return num[middle_index]
    else :
        middle_index_1 = int((len(num)/2))
        middle_index_2 = int((len(num)/2) - 1)
        return int(getMean([num[middle_index_1],num[middle_index_2]]))

def getstandardDeviation(num):
    if len(num) == 0:
        return 0
    else:
        mean = getMean(num)
        std_dev = 0
        for i in num:
            std_dev += (i - mean)**2
        return (std_dev/len(num))**0.5

def Variance(num):
    return getstandardDeviation(num)**2

def getStandardization(num):
    st = 0
    mean = getMean(num)
    st_dev = getstandardDeviation(num)
    for i in num:
        st += i - mean
    return st/st_dev

def minmaxstandardization(num):
    result = 0
    if len(num) == 0:
        return result
    else:
        min_val = min(num)
        max_val = max(num)
        for i in num:
            result = ((i-min_val)/(max_val - min_val))
        return result

..
```

